IS 300 — Lecture 4

◆ How is software categorized?

◆ What are the major functions of an operating system?

◆ How do the major operating systems compare?

◆ What programming language options exist and what are the implications relating to these choices?
How is software categorized?

Software

- Systems Software
- Applications Software

  - Horizontal (general purpose)
  - Vertical (special purpose)
What are the major functions of an operating system?

Supporting the GUI – Graphical User Interface

Managing system memory

Virtual Memory
- better utilization of RAM
- larger programs run in available RAM
- can hurt performance

Managing processing tasks

Multitasking (multiprogramming)
- improved productivity for user
- better utilization of resources

Multithreading

File Management

Network support

Single- versus multi-user systems

Enforcing access/security rules

Multiprocessing
How do the major operating systems compare?

UNIX

Advantages
- runs on many machines (portable)
- good network support
- powerful for the power user

Disadvantages
- can be cryptic (although good GUI are available)
  ```
  awk '/hello/,/goodbye/ {print}' filex
  ```

Linux ("Lean - icks")
- open standard
- very inexpensive

MAC OS (OS 9 and OS X)

Advantages
- nice GUI
- good network support
- architecturally sound
- protected memory
- preemptive multitasking

Disadvantages
- market share
Windows 95/98
Advantages
• nice GUI
• market share
• blends old/new technology

Disadvantages
• blends old/new technologies (FAT16/FAT32)
• memory not protected

Windows NT 4.0
Advantages
• nice GUI
• new technology (protected memory, preemptive multitasking)
• good network/security

Disadvantages
• market confusion
• complexity (resource use)

Windows 2000 Professional
Mobile (plug 'n play)
Internet enabled (ala Windows 98)
Resource requirements
• 133 MHz Pentium
• 64 MB RAM (128 MB better)
• 2 GB disk space

Windows 2000 Server
Supports a “Business Internet”
Resource requirements
• 133 MHz Pentium
• 256 MB RAM (more is better)
• 2 GB disk space
What programming language options exist and what are the implications relating to these choices?

<table>
<thead>
<tr>
<th></th>
<th>Low level (machine/assembly)</th>
<th>Procedure-oriented (BASIC, C)</th>
<th>Nonprocedural</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU dependent</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Solution orientation</strong></td>
<td>How</td>
<td>How</td>
<td>What</td>
</tr>
<tr>
<td><strong>Steps for computing the average:</strong></td>
<td></td>
<td></td>
<td>Compute average age</td>
</tr>
<tr>
<td>1. Sum ages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Count number of ages</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Divide sum by count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Programming skill level</strong></td>
<td>Very high</td>
<td>Moderate to high</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Machine efficiency</strong></td>
<td>Excellent</td>
<td>Moderate to good</td>
<td>Moderate to poor</td>
</tr>
</tbody>
</table>